

Desarrollo de una Base de Datos Nativa XML

Luis Fernando Espino Barrios
Instituto Tecnológico de Costa Rica
luisespino@yahoo.com
Noviembre 2009

Resumen: En este artículo se tratan elementos conceptuales alrededor del tema de bases de datos XML, tales como estructura de documentos, esquemas, persistencia y consultas. Además de presentar el desarrollo de una herramienta de bases de datos nativa XML con soporte de consultas tipo XPath en PHP. Dentro de la persistencia de los documentos XML hay dos enfoques, uno llamado XML nativo y otro XML-enabled, este último transforma la estructura jerárquica a un esquema relacional para su posterior almacenamiento. Se hace énfasis en el enfoque de XML nativo, debido al tiempo de respuesta, facilidad de uso y herramientas que lo soportan, como el *Document Object Model* (DOM). Además se describen dos lenguajes de consultas: XPath y XQuery, seleccionando XPath como lenguaje para la herramienta descrita por la simplicidad y soporte que posee.

Palabras clave: Bases de datos XML, *Document Object Model*, XML nativo, XML-enabled, XPath, XQuery.

1. Introducción

XML significa Lenguaje de Marcas Extensible, del inglés *eXtensible Markup Language*, es un conjunto de reglas para definir etiquetas semánticas que dividen un documento en partes e identifica las diferentes partes del documento [1]. XML está derivado de un lenguaje para estructurar documentos [2], conocido como lenguaje estándar generalizado de marcas, del inglés *Standard Generalized Markup Language*, SGML publicado en 1986.

El término marca se refiere a cualquier elemento en un documento del que no se tiene intención que sea parte de la salida impresa.

Un lenguaje de marcas es una descripción formal de la parte del documento que es el contenido, la parte que es marca y lo que significa la marca.

La especificación de XML 1.0 define los componentes de documentos XML individuales, particionados en estructuras lógicas y físicas.

XML expresa la información utilizando cuatro componentes que son las etiquetas, los atributos, los elementos y la jerarquía implícita. Tal como se menciona en [3], estos componentes ayudan a un único propósito, el cual, es representar diferentes dimensiones de la información.

La principal desventaja radica en que los nombres de etiquetas se repiten por todo el documento.

Entre las ventajas se mencionan que: las etiquetas hacen que el mensaje sea autodocumentado, el formato del documento no es rígido y que permite estructuras anidadas.

En [3], se mencionan tres capacidades importantes: la heterogeneidad, en donde cada registro puede contener diferentes campos de datos, similar al mundo real; la extensibilidad, en donde nuevos tipos de datos pueden ser agregados; y la flexibilidad, en donde los campos pueden cambiar de tamaño y configuración de instancia a instancia.

XML se ha utilizado como un lenguaje de facto para la exportación e intercambio de datos en internet, sin embargo, su utilización no está limitada para ser un formato de ese tipo, porque también existen aplicaciones que utilizan XML tales como: la administración de conocimiento en la bioinformática, la administración de datos, el intercambio de datos geográficos, las aplicaciones orientadas a la exploración del espacio, el marco para bases de datos inductivas y recientemente estudios para utilizar XML *Data Warehouse*.

En el resto del artículo se tratará de una manera conceptual los temas de esquemas de documentos, almacenamiento, consultas, sintaxis y por último datos técnicos de la implementación de un sistema de bases de datos nativa XML.

2. Esquema de documentos XML

La definición [2] de tipos de documentos DTD es una parte opcional de un documento XML.

El propósito es similar a un esquema, restringir el tipo de información.

Entre algunas limitaciones de DTD están:

- No se puede declarar el tipo de cada elemento y de cada atributo.
- Es difícil utilizar el mecanismo de DTD.
- Hay cierta falta de tipos.

XML Schema es un intento de reparar las deficiencias de DTD, siendo un esquema más sofisticado.

El esquema XML es un lenguaje que describe la estructura y restricciones de los documentos XML, que mejora los problemas de DTD o definición de tipo de documentos. En otro sentido representa los metadatos para un documento XML asociado o clases de documentos XML [4].

Como parte de una correcta descripción de documentos XML es necesario tener en cuenta ciertos errores comunes [3] que se cometen al crear documentos XML:

- Contexto inadecuado describiendo elementos de datos (uso incompleto de etiquetas).
- Instrucciones inadecuadas de cómo interpretar los datos (uso incompleto de atributos).
- Utilización de atributos como elementos (uso inapropiado de atributos).
- Uso de elementos de datos como metadatos.
- Redundancia de etiquetas.
- Atributos que no describen a los elementos.

3. Persistencia de documentos XML

Existen dos tipos de almacenamiento:

- XML Nativo: Utiliza un modelo de almacenamiento basado en los documentos XML, el formato puede depender del propietario y no es necesario que se almacene en archivos de texto.
- XML-enabled: Es mapear un XML a una base de datos relacional, aceptando un XML como entrada y haciendo una transformación para hacer el XML como salida.

En [5] se define una base de datos XML como una colección de documentos XML y sus partes, mantenidas por un sistema que tiene capacidades para manejar y controlar la colección misma, y la información representada por esa colección.

Las bases de datos XML tiene sus raíces en las bases de datos jerárquicas y textuales. Por un lado las bases de datos jerárquicas [6] se componen de un conjunto ordenado de árboles, formado por múltiples ocurrencias de un solo tipo de árbol. Los tipos de arboles consisten es un solo tipo de registro raíz con más arboles dependientes. Por otro lado las bases de datos textuales describen un modelo de datos como una tabla dentro de un archivo de texto, generalmente cada registro se delimita por una línea o por un carácter especial. Muy utilizado en ambientes Unix.

3.1. Representación relacional de XML

Por la estructura tipo árbol que presentan los documentos XML es de manera sistemática representarlo como un esquema relacional.

Esto se realiza por la asociación de uno a muchos entre cada nodo, claramente la estructura arbórea creará documentos des-normalizados.

En [7] se muestra un ejemplo sencillo de correspondencia que permite ver la representación relacional:

Se supone el XML siguiente:

```
<company id="c1">
  <section id="s1">
    <employee id="e1"/>
    <employee id="e2"/>
  </section>
  <section id="s2">
    <employee id="e3"/>
  </section>
</company>
```

El esquema relacional quedaría de la siguiente manera:

company	section	employee
c1	s1	e1
c1	s1	e2
c1	s2	e3

Tabla 1: Esquema relacional equivalente

4. Consultas para documentos XML

Herramientas son necesarias para extraer la información de documentos XML, análogamente, una consulta relacional extrae una relación, una consulta de XML extrae XML.

Hay dos lenguajes definidos como estándares para extracción de información en XML:

- XPath y
- XQuery

4.1. XPath

XPath trata partes de los documentos XML mediante expresiones de rutas de acceso. Estas rutas están basadas en las rutas de acceso de las bases de datos orientadas a objetos.

La versión actual es la 2.0, en [8] se menciona que en la especificación se supone un entendimiento básico de XPath 1.0, según la definición en ambas versiones [9] y [10] XPath es un lenguaje para hacer frente a las partes de un documento XML.

La diferencia radica que la versión 2.0 es un lenguaje más poderoso que opera en dominios grandes de tipos de datos y está orientado al procesamiento de secuencias.

4.2. XQuery

XQuery es otro lenguaje de consulta, procede de un lenguaje llamado Quilt basado en XPath.

Las consultas de XQuery difieren de SQL. Existen diferentes instrucciones dentro de las expresiones FLWOR:

- For,
- Let,
- Where,
- Order by, y
- Return.

Según la W3C, XQuery [11] está diseñado para ser un lenguaje en que las consultas son concisas y de fácil comprensión, también es flexible como para consultas de un amplio espectro de fuentes de información XML, incluyendo bases de datos y documentos.

4.3. Algunas equivalencias entre XPath y XQuery

Excluyendo otros elementos de XQuery como las funciones y el ordenamiento, básicamente cualquier consulta realizada en XQuery tiene una consulta equivalente en XPath.

Por ejemplo, una consulta XQuery bajo las expresiones FLWOR sería:

```
for $a in /catalogo/cd
let $titulo:=$a/titulo
where $a/precio > 10
return $titulo
```

Su equivalente en XPath sería:

```
/catalogo/cd[precio>10]/titulo
```

5. Sintaxis básica de XPath

5.1. Selección

- **nodename:** Selecciona todos los nodos hijos que tengan ese nombre.
- **/:** Selecciona desde el nodo raíz.
- **//:** Selecciona los nodos que contengan el nombre del nodo no importando su ubicación.
- **..:** Selecciona el nodo actual.
- **...:** Selecciona el padre del nodo actual
- **@:** Selecciona atributos.

5.2. Predicados

- **[x]:** Selecciona el x nodo.
- **[last()]:** Selecciona el último nodo.
- **[@name]:** Selecciona los nodos que contenga un atributo llamado name.

- [elemento>x]: Selecciona los nodos que tengan un valor en el elemento indicado mayor que x.
- *: Selecciona todos los nodos.

5.3. Operadores

- |: Realiza la unión de dos consultas, en el orden en que aparecen en el documento.
- +,-,*,div: Operadores aritméticos.
- =, !=, <, >, >=, <=: Operadores lógicos

6. Implementación

Se implementó una herramienta de software orientada a web que funciona como una base de datos nativa con soporte de consultas de tipo XPath.

Los lenguajes de programación utilizados son PHP y JavaScript.

Para lograr alcanzar el objetivo se requirió de dos bibliotecas:

- CodeMirror¹: Es una biblioteca de JavaScript que puede ser utilizada para crear un editor para contenido de código fuente.
- DOM²: Es el *Document Object Model* para PHP, más que una biblioteca es una extensión que permite operar documentos XML a través del API con PHP 5.

Para lograr un cuadro de diálogo en Javascript sin problema con los navegadores se elaboró una tabla para mostrarla encima de la ventana actual.

La llamada de la función se hace de la siguiente manera:

```
<a href="" onclick="DialogBox('Type the file name:', new_xml); return false">New XML</a>
```

Las líneas en JavaScript importantes son:

```
var dialogBox = document.createElement('div');
...
dialogBox.innerHTML = dialogBoxContents;
...
document.body.appendChild(dialogBox);
```

La primera crea dinámicamente un nuevo div, en este se escribe el HTML que se desea, en este caso sería el código de la forma de HTML y por último se agrega a la ventana como hijo.

El código para subir el documento XML al servidor en PHP es el siguiente:

```
$location = "xml/" . $_FILES["file"]["name"];
if (move_uploaded_file($_FILES["file"]
    ["tmp_name"], $location))
    $filename = $_FILES["file"]["name"];
else
    echo "Upload error!";
```

El código para abrir un XML en PHP es:

```
$fh = fopen($location, 'r');
$data = fread($fh, filesize($location));
fclose($fh);
```

El código que recibe el XML y el método para ejecutar la consulta XPath en PHP es:

```
$location = $_REQUEST["location"];
$xml = new DOMDocument();
$xml->load($location);
$xmlpath = new DOMXPath($xml);
$elements = $xmlpath->evaluate(stripslashes
    ($_REQUEST["query"]));
```

¹ <http://marijn.haverbeke.nl/codemirror/>

² <http://www.php.net/manual/en/book.dom.php>

7. Manual de usuario

Un prototipo de la herramienta se encuentra en <http://nxdb.comlu.com>, a continuación se muestra la Figura 1: Página principal del sitio:



Home - [New XML](#) [Open XML](#) [Upload XML](#)

Figura 1: Página principal del sitio

Las opciones disponibles son:

- Nuevo XML: Crea un nuevo documento en el servidor.

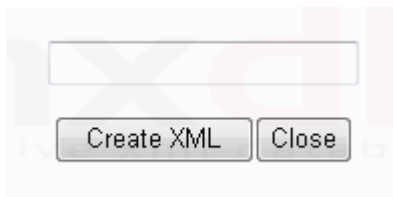


Figura 2: Crear XML

- Abrir XML: Selecciona un archivo que previamente haya sido almacenado en el servidor.

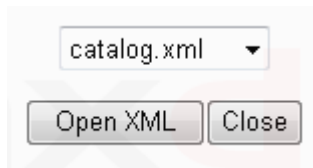


Figura 3: Abrir XML

- Subir XML: Sube un documento XML del usuario hacia el servidor.

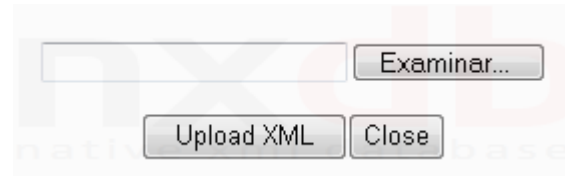


Figura 4: Subir XML

Con cualquier opción seleccionada, se ingresa a la página de edición.



Figura 5: Área de edición de XML

Hay dos opciones en la edición:

- Guardar el documento y
- Bajar el documento guardado.

Con respecto a las consultas, hay un botón que ejecuta la consulta, y tiene tres tipos de respuesta:

- La respuesta si la consulta está correcta.
- La respuesta de que la consulta no tuvo resultados.
- La respuesta de que la consulta tiene un problema de sintaxis.

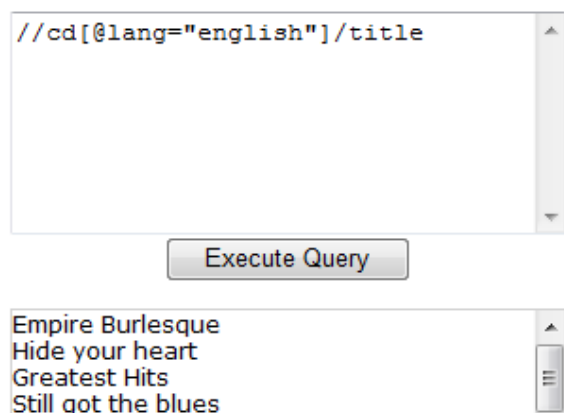


Figura 6: Área para consultar en XML

8. Conclusiones

XML es un lenguaje de marcas que de facto se utiliza como lenguaje de exportación e intercambio, pero tiene muchas aplicaciones adicionales de uso.

Recientemente se ha utilizado para crear las bases de datos XML con dos enfoques de persistencia, uno nativo y otro habilitado por un sistema de bases de datos relacional.

Para realizar consultas en documentos XML hay dos lenguajes: XPath y XQuery, que por simplicidad y soporte en PHP se utilizó XPath para la herramienta presentada.

Se presentó una herramienta orientada a web que maneja una base de datos nativa XML, desarrollada en PHP, llamada NXDB y tiene soporte para la edición de documentos XML y para consultas tipo XPath.

9. Referencias bibliográficas

[1] E. Harold, *XML 1.1 Bible*, 3rd ed. United States of America: Wiley Publishing, Inc., 2004.

[2] A. Silberschatz, H. Korth, and S. Sudarshan, *Fundamentos de Bases de Datos*, 5th ed. España: McGraw-Hill, 2006.

[3] A. B. Chaudhri, A. Rashid, and R. Zicari, *XML Data Management: Native XML and XML-Enabled Database Systems*. United States of America: Addison Wesley, 2003.

[4] C. Campbell, A. Eisenberg, and J. Melton, "XML Schema," in *SIGMOD Record*, Vol. 32, No. 2, 2003, pp. 86-101.

[5] A. Salminen and F. Tompa, "Requirements for XML Document Database Systems," in *Proceedings of the 2001 ACM Symposium on Document Engineering*, New York, USA., 2001, pp. 85-94.

[6] C. J. Date, *Introducción a los Sistemas de Bases de Datos*, 5th ed. México: Addison Wesley Iberoamericana, 1992.

[7] T. Saito and S. Morishita, "Relational-Style XML Query," in *SIGMOD'08*, Vancouver, BC, Canada., 2008, pp. 303-314.

[8] E. Lenz, "What's New in XPath 2.0," O'Reilly Media, Inc, 2002.

[9] W3C (MIT, INRIA, Keio), "XML Path Language (XPath) Version 1.0," W3C, 1999.

[10] W3C (MIT, ERCIM, Keio), "XML Path Language (XPath) 2.0," W3C, 2007.

[11] W3C (MIT, ERCIM, Keio), "XQuery 1.0: An XML Query Language," W3C, 2007.